Computer Science

34^{th} Annual High School Programming Contest

Sponsored by transfinder

April 8, 2022

Gold Problem #1: Surface Area

Background Information: The formula for the surface area *A* of a right rectangular prism, with a width *w*, a length *l*, and a height *h*, is

$$A = 2(wl + hl + hw)$$

Your program will be given three positive integers, each less than one thousand. The three integers will represent the width, length, and height of a rectangular prism. Your program will then calculate and print out the prism's surface area.

Programming Problem:

Input: Three positive integers, each less than 1,000. All inputs will be on separate lines.
Output: The corresponding surface area value.

Example 1: Input:
```
6
3
2
```
Output:
```
72
```

Example 2: Input:
```
4
10
7
```
Output:
```
276
```

Example 3: Input:
```
1
5
2
```
Output:
```
34
```

34th Annual High School Programming Contest

Sponsored by transfinder

April 8, 2022

Gold Problem #2: Stratego

Background Information: In the classic board game Stratego, you have various pieces, most of which represent soldiers; there are also BOMBs and a FLAG.

Here are the soldiers in rank order from highest to lowest:

MARSHAL
GENERAL
COLONEL
MAJOR
CAPTAIN
LIEUTENANT
SERGEANT
MINER
SCOUT
SPY

In the game, a soldier may attack any other defending piece, resulting in one or both pieces being removed according to the following rules:

- If the FLAG is attacked, it is always removed.
- Any soldier other than a MINER attacking a BOMB is removed. When a MINER attacks a BOMB, the BOMB is removed.
- If a SPY attacks a MARSHAL (but not vice versa), then the MARSHAL is removed.
- If a soldier attacks a soldier of the same rank, both pieces are removed.
- In all other cases, the lower-ranking piece is removed.

Your program will read in two legal Stratego pieces: an attacking piece first and then a defending piece. Your program will then print out which piece(s) are removed, according to the stated rules.

Programming Problem:

Input: An attacking piece string and a defending piece string on separate lines.
Output: The piece that is removed, in the form `<NAME> REMOVED` (all caps, one space of separation). If both pieces are removed, output `BOTH REMOVED`.

Example 1: Input:
```
SERGEANT
CAPTAIN
```
Output:
```
SERGEANT REMOVED
```

Example 2:	Input:	`SPY` `MARSHAL`
	Output:	`MARSHAL REMOVED`
Example 2:	Input:	`COLONEL` `BOMB`
	Output:	`COLONEL REMOVED`
Example 3:	Input:	`MINER` `BOMB`
	Output:	`BOMB REMOVED`
Example 4:	Input:	`MARSHAL` `SPY`
	Output:	`SPY REMOVED`
Example 5:	Input:	`GENERAL` `GENERAL`
	Output:	`BOTH REMOVED`

Computer Science

34^{th} Annual High School Programming Contest

Sponsored by transfinder

April 8, 2022

Gold Problem #3: Repeating Binary Fractions

Background Information: In math class, a repeating decimal pattern can be represented as a rational number in lowest terms *a/b*, where *a*, *b* are integers and *a* is non-zero. For example the repeating decimal 0.36363636… is equal to 36/99, or 4/11 in lowest terms. The principle is the same with repeating binary patterns (patterns in base 2 only consisting of zeros and ones).

Your program will be given the repeating binary pattern immediately following the binary point of a number. It will output the fractional, lowest terms, equivalent in base-10.

Programming Problem:

Input: A sequence of 0's and 1's that represent the repeated portion of a binary fraction up to 30 binary digits.
Output: The equivalent base-10 fraction in lowest terms, written in A/B format.

Example 1: Input:
```
0011
```
Output:
```
1/5
```

Example 2: Input:
```
01
```
Output:
```
1/3
```

Example 3: Input:
```
0000000000011110
```
Output:
```
2/4369
```

Computer Science

34th Annual High School Programming Contest

Sponsored by transfinder

April 8, 2022

Gold Problem #4: Wordle

Background Information: Wordle is a popular word guessing game, in which you attempt to guess a hidden five letter word (called the target) in at most six attempts. Each attempt will result in a colorization of your guess, according to the following scheme:

- Letters that are not in the word will turn dark.
- Letters that are in the word and are in the correct spot in the target will turn green.
- Letters that are in the word but are in a different spot will turn yellow. If $K \geq 2$ or more letters that are the same in a guess appear in the target $T < K$ times, then only the leftmost T letter(s) will turn yellow; the rest will turn dark. For example, if the target word is IDIOM, and the guess is DADDY, then the coloring of DADDY will be yellow for the first "D", and then dark for the remaining letters.

Your program will read in a target word (the answer) followed by an integer $1 \leq N \leq 6$ representing the number of guesses, followed by N guess words. Each word will be exactly 5 uppercase letters.
Your program will then output N strings consisting of the letters G, Y, and D for each guess, based upon the colorization scheme noted above. The letter G represents green, Y represents yellow, and D represents dark.

Programming Problem:

Input: 1 5-letter word, followed by an integer N in [1, 6], followed by N 5-letter words, each on separate lines, all in uppercase letters
Output: N 5-letter output strings made up of G's Y's and D's for each of the N guesses in order.

Example 1:

Input:	Output:
BATON	YDDDY
4	YYYDD
AUDIO	YYYDY
TOADS	GGGGG
ABOUT	
BATON	

Example 2: Input:

```
ICING
6
ONION
ANION
MIMIC
GOING
COMIC
ABOUT
```

Output:

```
DYGDD
DYGDD
DYDYY
DDGGG
YDDYD
DDDDD
```

Example 3: Input:

```
MIGHT
6
OTTER
TIGHT
SIGHT
LIGHT
NIGHT
FIGHT
```

Output:

```
DYDDD
DGGGG
DGGGG
DGGGG
DGGGG
DGGGG
```

Computer Science

34th Annual High School Programming Contest

Sponsored by transfinder

April 8, 2022

Gold Problem #5: Clothing Restrictions

Background Information: You are about to go to school for the day. However, you have many fashion options from which to choose. You must wear a HAT, a SHIRT, PANTS, and SOCKS. Each item has four separate colors from which to choose: GREEN, RED, BLUE, and YELLOW. How many different choices do you have? Without restrictions, you have 4x4x4x4 = 256 options. However, how many do you have if you have restrictions?

Your program will be given *n* distinct restrictions to your wardrobe choices. A restriction is a colored clothing item not being worn with another clothing item. For example, a possible restriction is

GREEN HAT YELLOW SOCKS

This means that you cannot wear a GREEN HAT with YELLOW SOCKS, thus slightly limiting your choices. All restrictions will involve the listed colors and apparel.

Programming Problem:

Input: an integer *n*, followed by *n* clothing restrictions of the previous form, each on a separate line

Output: The number of possible combinations that remain

Example 1: Input:
```
0
```
Output:
```
256
```

Example 2: Input:
```
1
BLUE HAT YELLOW SOCKS
```
Output:
```
240
```

Example 3: Input:
```
2
BLUE HAT YELLOW SOCKS
GREEN PANTS BLUE SHIRT
```
Output:
```
225
```

Hints on reading this input:

As you can see, you will need to read four words that are on the same line of input into variables. In case this is not something you have done before, we give examples on the next page.

In Python, the following will read a line such as `BLUE HAT YELLOW SOCKS` and place those 4 words into the variables `color1`, `item1`, `color2`, and `item2`, respectively:

```
(color1, item1, color2, item2) = input().split()
```

In Java, if you have a Scanner named `s`, the 4 words could be placed into variables `color1`, `item1`, `color2`, and `item2` with these lines:

```
String color1 = s.next();
String item1 = s.next();
String color2 = s.next();
String item2 = s.next();
```

Computer Science

34th Annual High School Programming Contest

Sponsored by transfinder

April 8, 2022

Gold Problem #6: The Transfinder Problem

Background Information: You are in control of a busing system, where you have $n <= 10$ bus routes that you can use to travel to stations numbered from 1 to $p <= 1000$. You are always starting at station number 1. You are given the bus schedule for the day, which is a list of station numbers that the buses on each route will stop to pick up and drop off passengers, and the ordering which they do so. Your task is to determine the number of different ways you can travel from station 1 to station *p*.

For example, on a trip to from station 1 to station 10, you have

- One bus route that stops at every station
- One bus route that stops at stations 1, 4, and 10

There are 4 ways to do this. The first two are simple: you can travel on the first or second bus route from 1 all the way to 10. The other two involve transfers. You could start on the first bus route, take it to station 4, then transfer from the first bus route to the second bus route at station 4, and take the second bus route to station 10. You could start on the second bus route, take it to station 4, then transfer from the second bus route to the first bus route at station 4, and take the first bus route to station 10.

Note that each bus route does not have to stop at station 1, nor does each bus route have to stop at station *p*. If we were to have one bus route that stops only at 1, 5, and 10, and another that had stops only at station 6 and station 9, the second route would not be useful for our travels, since we must start on the route that has a stop at 1, and we have no way to get to station 6 to switch to the the other bus route.

Programming Problem:

Input: The destination station number *p*, followed by the number of bus routes *n*, on separate lines. Then, you will read in *n* pairs of lines, the first in each pair being the number of stops *k*, and the next line being *k* increasing numbers on the interval [1, *p*] representing the stations at which this bus route has stops.

Output: The number of different ways you can travel from station 1 to station *p*.

Example 1: Input:

```
10
2
10
1 2 3 4 5 6 7 8 9 10
3
1 4 10
```

Output:

```
4
```

Example 2: Input:

```
10
2
3
1 5 10
2
6 9
```

Output:

```
1
```

Example 3: Input:

```
13
3
13
1 2 3 4 5 6 7 8 9 10 11 12 13
7
1 3 5 7 9 11 13
4
1 5 9 13
```

Output:

```
125
```

Example 4: Input:

```
8
2
3
1 4 7
3
2 5 8
```

Output:

```
0
```

Some hints on reading the input for this problem are below.

<u>Hints on reading this input:</u>

As you can see, you will need to read a line with several numbers that are on the same line of input into variables (likely an array). In case this is not something you have done before, we give some tips below.

In Python, this function will read a line of input known to contain 1 or more space-separated integer values and will return an array of those values as integers:

```
def input_to_intarray():
    arr = input().split()
    for i in range(len(arr)):
        arr[i] = int(arr[i])
    return arr
```

Using this, if you would like to read the next line of input into a variable `a`, you could write:

```
a = input_to_intarray()
```

In Java, we are more likely to take advantage of the previous line, which tells us how many numbers to expect on the line of station numbers. Say we read the number of stations into a variable `n`, we could then use the following to read in the numbers from a Scanner `s`, into an array `a`:

```
int a[] = new int[n];
for (int i = 0; i < n; i++) {
    a[i] = s.nextInt();
}
```

Computer Science

34^{th} Annual High School Programming Contest

Sponsored by transfinder

April 8, 2022

Gold Problem #7: Postage Stamps

Background Information: Suppose you can purchase a total of *n* postage stamps, but can only have them in two distinct integer denominations, a_1 and a_2. What is the highest postage amount, *h*, such that you can represent every positive integer amount 1 through *h* using a total of at most *n* stamps of denominations a_1 and a_2? Well, it definitely depends on which values you choose for a_1 and a_2!

Your program will take as input the value *n*, and it will output the highest number *h*, such that the optimal choices for stamp values a_1 and a_2 will produce combinations of at most *n* stamps that add up to all integers on the interval [1, *h*]. You will also output the stamp values a_1 and a_2, where $a_1 < a_2$, which can be used to produce those postage values on the interval [1, *h*].

Programming Problem:

Input: The maximum number of stamps that can be used, $n <= 75$.
Output: The highest number, *h*, such that each total of stamp values [1, *h*] can be produced. The stamp values a_1 and a_2, both integers are also outputted on the next line.

Example 1: Input:
```
1
```
Output:
```
2
1 2
```

Example 2: Input:
```
3
```
Output:
```
7
1 3
```

Example 3: Input:
```
5
```
Output:
```
14
1 4
```

Example 4: Input:
```
7
```
Output:
```
23
1 5
```